C-Log: A Logic of Causality

Bart Bogaerts *Joost Vennekens* *Marc Denecker*
Jan Van den Bussche

Report CW 656, Februari 2014

KU Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

C-Log: A Logic of Causality

Bart Bogaerts *Joost Vennekens* *Marc Denecker*
Jan Van den Bussche

Report CW 656, Februari 2014

Department of Computer Science, KU Leuven

Abstract

Cause-effect relations are an important part of human knowledge. In real life, humans often reason about complex causes linked to complex effects. By comparison, existing formalisms for representing knowledge about causal relations are quite limited in the kind of specifications of causes and effects they allow. In this paper, we present the new language C-Log, which offers a significantly more expressive representation of effects, including such features as the creation of new objects. We show how Approximation Fixpoint Theory can be used to define a formal semantics that captures the intuitions underlying this language. We also compare C-Log with several related languages and paradigms, including inductive definitions, disjunctive logic programming, business rules and extensions of Datalog.

MSC : Primary : I.2.4

C-log: A Logic of Causality

Bart Bogaerts and **Joost Vennekens** and **Marc Denecker**
Department of Computer Science, KU Leuven
{bart.bogaerts, joost.vennekens, marc.denecker}@cs.kuleuven.be

Jan Van den Bussche
Hasselt University & transnational University of Limburg
jan.vandenbussche@uhasselt.be

Abstract

Cause-effect relations are an important part of human knowledge. In real life, humans often reason about complex causes linked to complex effects. By comparison, existing formalisms for representing knowledge about causal relations are quite limited in the kind of specifications of causes and effects they allow. In this paper, we present the new language C-log, which offers a significantly more expressive representation of effects, including such features as the creation of new objects. We show how Approximation Fixpoint Theory can be used to define a formal semantics that captures the intuitions underlying this language. We also compare C-log with several related languages and paradigms, including inductive definitions, disjunctive logic programming, business rules and extensions of Datalog.

1 Introduction

Cause-effect relations are an important part of human knowledge. There exist a number of knowledge representation languages (McCain and Turner 1996; Vennekens, Denecker, and Bruynooghe 2009; Cabalar 2012) in which logic programming style rules are used to represent such relations. The basic idea in all these approaches is that the head of such a rule represents an effect that is caused by its body. In this paper, we are particularly concerned with CP-logic (Vennekens, Denecker, and Bruynooghe 2009). More specifically, we consider the variant of CP-logic without probabilities, and we will extend this language with three features: dynamic non-determinsitic choice; object creation; and recursive nesting of cause-effect relations. We call the resulting language C-log. The main technical contribution of this paper is to show how approximation fixpoint theory can be used to provide a clean semantics for recursive causal theories expressed in C-log.

Let us begin by recalling the guiding principles behind CP-logic. When compared to predecessors, such as the causal logic of (McCain and Turner 1996), one of the important contributions of this languages is to add two modelling principles that are common in causal modelling. The first is the distinction between *endogenous* and *exogenous* properties, i.e., those whose value is determined by the causal laws in the model and those whose value is not, respectively (Pearl 2000). The second is *default-deviant* assumption, used also by, e.g., (Hall 2004; Hitchcock 2007). The idea here is to assume that each endogenous property of the domain has some "natural" state, that it will be in whenever nothing is acting upon it. For ease of notation, CP-logic identifies the default state with falsity, and the deviant state with truth. For example, consider the following simplified model of a bicycle, in which a pair of gear wheels can be put in motion by pedalling:

$$\mathrm{Turn(BigGear)} \leftarrow \mathrm{Pedal}. \quad (1)$$

$$\mathrm{Turn(BigGear)} \leftarrow \mathrm{Turn(SmallGear)}. \quad (2)$$

$$\mathrm{Turn(SmallGear)} \leftarrow \mathrm{Turn(BigGear)}. \quad (3)$$

Here, Pedal is exogenous, while $\mathrm{Turn(BigGear)}$ and $\mathrm{Turn(SmallGear)}$ are endogenous. The semantics of this causal model is given by a straightforward "execution" of the rules. The domain starts out in an initial state, in which all endogenous atoms have their default value *false* and the exogenous atom Pedal has some fixed value. If Pedal is true, then the first rule is applicable and may be fired ("Pedal causes $\mathrm{Turn(BigGear)}$") to produce a new state of the domain in which $\mathrm{Turn(BigGear)}$ now has its deviant value *true*. In this way, we construct the following sequence of states (we abbreviate symbols by their first letter):

$$\{\mathrm{P}\} \overset{(1)}{\rightarrow} \{\mathrm{P}, \mathrm{T(B)}\} \overset{(3)}{\rightarrow} \{\mathrm{P}, \mathrm{T(B)}, \mathrm{T(S)}\} \overset{(2)}{\rightarrow} \{\mathrm{P}, \mathrm{T(B)}, \mathrm{T(S)}\} \quad (4)$$

Note that firing rule (2) does not change the state of the world, because its effect is already true. Moreover, it is obvious that this will always be the case, so this rule may seem redundant. However, many interesting applications of causal models require the use of interventions (Pearl 2000), e.g., to evaluate counterfactuals or to predict the effects of actions. As shown in (Vennekens, Denecker, and Bruynooghe 2010), rule (2) allows CP-logic to represent this example in a way that produces the correct results for all conceivable interventions in a manner that is more modular and more concise than, among others, Pearl's structural models (Pearl 2000).

After rules (1), (3) and (2) have all fired, there are no more rules left whose body is satisfied and that have not yet fired. At this point, the process is at an end and the domain has reached a final state. It is this final state, rather than the details of the intermediate process, that we are really interested

in. One of the most important properties of CP-logic is that, while there may be any number of different processes derived from a causal theory, the final state that is eventually reached is unique for any given interpretation for the exogenous predicates—at least, for examples such as this one. In general, CP-logic also allows rules with a *non-deterministic* effect, such as:

$$(\mathrm{Turn}(\mathrm{SmallGear}) : 0.99)\ \mathbf{Or}\ (\mathrm{ChainBreaks} : 0.01) \leftarrow \mathrm{Turn}(\mathrm{BigGear}).$$

Now, the cause $\mathrm{Turn}(\mathrm{BigGear})$ produces one of two possible effects, and there is an associated probability distribution over these two possibilities. The effect on the semantics is that, instead of a linear progression of states as in (4), we get a tree structure in which each firing of a non-deterministic rule introduces a branching of possibilities. When considering also the probabilities associated to non-deterministic choices, the tree defines a probability distribution over its leaves, i.e., over the final states that may be reached. It was shown in (Vennekens, Denecker, and Bruynooghe 2009) that, given a specific interpretation for the exogenous atoms, this distribution is unique, even though there may exists many probability trees that produce it.

In many circumstances, the precise values of the probabilities are not of interest. In such cases, a non-probabilistic variant of CP-logic may be used, in which these are omitted. The head of a rule is then simply a disjunction:

$$\mathrm{Turn}(\mathrm{SmallGear})\ \mathbf{Or}\ \mathrm{ChainBreaks} \leftarrow \mathrm{Turn}(\mathrm{BigGear}).$$

The trees then no longer produce a probability distribution over final states, but simply describe the set of all final states that may be reached. In other words, this formalism has a possible world semantics. It is this non-probabilistic variant that concerns us in this paper.

Like other rule-based approaches to causality, CP-logic uses a very simple way of specifying the possible effects of some cause, namely, as a disjunction of ground atoms. Clearly, this does not—or, at least, not directly—cover many interesting phenomena that may occur in practice:

- A robot enters a room, opens some of the doors in this room, and then leaves by one of the doors that are open. The robot's leaving corresponds to a non-deterministic choice between a *dynamic* set of alternatives, which is determined by the robot's own actions, and therefore cannot be hard-coded into the head of a rule. A language construct for representing such choices is present in P-log (Baral, Gelfond, and Rushton 2004).
- A stallion and a mare that are put in the same field may cause the birth of a foal. Therefore, not only the properties of these horses are governed by causal laws, but also their very existence.
- A horse being the parent of a foal is itself a cause for its own height to have a causal link to the height of the foal. Therefore, causal laws may be nested, in the sense that an effect can itself again consist of an entire causal law.

The goal of this paper is to develop an expressive knowledge representation language that is able to represent these more complex effects, and others like them, in a direct way. Moreover, we want to do this in a way that extends the approach of CP-logic. To summarise, the formal semantics of the language should consist of a set of possible worlds, each of which can be constructed by a non-deterministic causal process. This process will take place in the context of a fixed interpretation for the exogenous atoms. It will start from an initial state in which each of the endogenous atoms is at its default value false. The causal laws of our language will then "fire" and flip atoms to their deviant value, until no more such flips are possible. Whereas in CP-logic these flips happen one atom at a time, our extended language will flip *sets* of atoms at the same time. Moreover, our logic will present syntax and semantics for object-creation, as is needed in the second of the above examples.

The rest of this paper is structured as follows: we start by introducing causal effect expressions (CEEs) and their informal semantics in Section 2. In Section 3, we formally define the effect set of a CEE in a structure (the set of atoms that is flipped in in one step of the process) and we use this notion to formalise the underlying causal process in the case where causality is monotone. Next, we start Section 4 with a discussion about why these techniques do not work in the general (non-monotonic) case, and suggest solutions based on Approximation Fixpoint Theory. We end this section with the semantics of causal theories in the general case. In Section 5, we explain how C-log can be integrated in FO. We conclude in Section 6 by comparing C-log with various other paradigms, including inductive definitions (Denecker and Ternovska 2008), disjunctive logic programming with existential quantifications (You, Zhang, and Zhang 2013), Business Rules systems (Business Rules Group 2000) and Datalog extensions (Green, Aref, and Karvounarakis 2012).

2 Syntax and Informal Semantics

We assume familiarity with basic concepts of first-order logic (FO). Vocabularies, formulas, and terms are defined as usual. A Σ-structure interprets all symbols (including variable symbols) in Σ; $D^{\mathcal{I}}$ denotes the domain of $\mathcal{I}$ and $\sigma^{\mathcal{I}}$, with σ a symbol in Σ, denotes the interpretation of σ in $\mathcal{I}$. We use $I[\sigma : v]$ for the structure $\mathcal{J}$ that equals $\mathcal{I}$, except on σ: $\sigma^{\mathcal{J}} = v$. *Domain atoms* are atoms of the form $P(\bar{d})$ where the d_i are domain elements. We use restricted quantifications (Preyer and Peter 2002), e.g., in FO, these are formulas of the form $\forall x[\psi] : \varphi$ or $\exists x[\psi] : \varphi$, meaning that φ holds for all (resp. for a) x such that ψ holds. The above expressions are syntactic sugar for $\forall x : \psi \Rightarrow \varphi$ and $\exists x : \psi \wedge \varphi$, but such a reduction is not possible for other restricted quantifiers that we will define below. We call ψ the *qualification* and φ the *assertion* of the restricted quantifications. From now on, let Σ be a relational vocabulary, i.e., Σ consists only of predicate, constant and variable symbols.

2.1 Syntax

Definition 2.1. Causal effect expressions *(CEE) are defined inductively as follows:*

- *if $P(\bar{t})$ is an atom, then $P(\bar{t})$ is a CEE,*

- *if φ is a first-order formula and C' is a CEE, then $C' \leftarrow \varphi$ is a CEE,*
- *if C_1 and C_2 are CEEs, then C_1 $\mathbf{And}$ C_2 is a CEE,*
- *if C_1 and C_2 are CEEs, then C_1 $\mathbf{Or}$ C_2 is a CEE,*
- *if x is a variable, φ is a first-order formula and C' is a CEE, then $\mathbf{All}\, x[\varphi] : C'$ is a CEE,*
- *if x is a variable, φ is a first-order formula and C' is a CEE, then $\mathbf{Select}\, x[\varphi] : C'$ is a CEE,*
- *if x is a variable and C' is a CEE, then $\mathbf{New}\, x : C'$ is a CEE.*

We call a CEE an *atom-expression* (respectively *rule-*, **And**-, **Or**-, **All**-, **Select**- or **New**-*expression*) if it is of the corresponding form. We call a predicate symbol P *endogenous* in C if P occurs as the symbol of a (possibly nested) atom-expression in C, i.e., if P occurs in C but not only in first-order formulas. All other symbols are called *exogenous* in C. This is a straightforward generalisation of the same notions in CP-logic. An occurrence of a variable x is *bound* in a CEE if it occurs in the scope of a quantification over that variable ($\forall x$, $\exists x$, $\mathbf{All}\, x$, $\mathbf{Select}\, x$, or $\mathbf{New}\, x$) and *free* otherwise. A variable is *free* in a CEE if it has free occurrences. A *causal theory*, or *C-log theory* is a CEE without free variables. By abuse of notation, we often represent a causal theory as a set of CEEs; the intended causal theory is the **And**-conjunction of these CEEs. We often use Δ for a causal theory and C, C', C_1 and C_2 for its subexpressions.

2.2 Informal Semantics of CEEs

A CEE is a description of a set of causal laws. In the context of a state of affairs—which we represent, as usual, by a structure—a CEE non-deterministically describes a set of effects, a set of events that take place and change the state of affairs. We call such a set the *effect set* of the CEE. From a CEE C, we can derive causal processes similar to (4); a causal process is a sequence of intermediate states, starting from the default state, such that, at each state, the effects described by C take place. The process ends if the effects no longer cause changes to the state. A structure is a model of a CEE if it is the final result of such a process. There are two kinds of effects that can be described by a CEE: 1) flipping an atom from its default to its deviant state and 2) creating a new domain element. We now explain in a compositional way what the effect set of a CEE is in a given state of affairs.

The effect of an atom-expressions A is that A is flipped to its deviant state. A conditional effect, i.e., a rule expression, causes the effect set of its head if its body is satisfied in the current state, and nothing otherwise. The effect set described by an **And**-expression is the union of the effect sets of its two subexpressions; an **All**-expression $\mathbf{All}\, x[\varphi] : C'$ causes the union of all effect sets of $C'(x)$ for those x's that satisfy φ. An expression C_1 $\mathbf{Or}$ C_2 non-deterministically causes either the effect set of C_1 or the effect set of C_2; a **Select**-expression $\mathbf{Select}\, x[\varphi] : C'$ causes the effect set of C' for a non-deterministically chosen x that satisfies φ. An object-creating CEE $\mathbf{New}\, x : C'$ causes the creation of a new domain element n and the effect set of $C'(n)$.

Example 2.2. American citizenship can be obtained in several ways. One way is passing the naturalisation test. Another way is by playing the "Green Card Lottery", where each year a number of lucky winners are randomly selected and granted American citizenship. We model this as follows:

$$\mathbf{All}\, p[\mathrm{Applied}(p) \wedge \mathrm{PassedTest}(p)] : \mathrm{American}(p)$$
$$(\mathbf{Select}\, p[\mathrm{Applied}(p)] : \mathrm{American}(p)) \leftarrow \mathrm{Lottery}.$$

The first CEE describes the "normal" way to become an American citizen; the second rule expresses that one winner is selected among everyone who applies. If $\mathcal{I}$ is a structure in which Lottery holds, due to the non-determinism, there are many possible effect sets of the above CEE, namely the sets $\{\mathrm{American}(p) \mid p \in D^{\mathcal{I}} \wedge p \in \mathrm{Applied}^{\mathcal{I}} \wedge \mathrm{PassedTest}(p)^{\mathcal{I}}\} \cup \{\mathrm{American}(d)\}$ for some $d \in \mathrm{Applied}^{\mathcal{I}}$. The two CEEs are considered independent: the winner could be one of the people that obtained it through standard application, as well as someone else.

Note that in the above, there is a great asymmetry between $\mathrm{Applied}(p)$, which occurs as a qualification of **Select**-expression, and $\mathrm{American}(p)$, which occurs as a caused atom. This means that the effect will never cause atoms of the form $\mathrm{Applied}(p)$, but only atoms of the form $\mathrm{American}(p)$. This is one of the cases where the qualification of an expression cannot simply be eliminated.

Example 2.3. Hitting the "send" button in your mail application causes the creation of a new package containing a specific mail. That package is put on a channel and will be received some (unknown) time later. As long as the package is not received, it stays on the channel. In C-log, we model this as follows:

$$\mathbf{All}\, m, t[\mathrm{Mail}(m) \wedge \mathrm{HitSend}(m,t)] : \mathbf{New}\, p :$$
$$\mathrm{Pack}(p)\ \mathbf{And}\ \mathrm{Cont}(p,m)\ \mathbf{And}\ \mathrm{OnCh}(p,t+1)\ \mathbf{And}$$
$$\mathbf{Select}\, d[d>0] : \mathrm{Received}(p,t+d)$$
$$\mathbf{All}\, p,t[\mathrm{Pack}(p) \wedge \mathrm{OnCh}(p,t) \wedge \neg\mathrm{Received}(p,t)] :$$
$$\mathrm{OnCh}(p,t+1)$$

Suppose an interpretation $\mathrm{HitSend}^{\mathcal{I}} = \{(\mathrm{MyMail}, 0)\}$ is given. A causal process then unfolds as follows: it starts in the initial state, where all endogenous predicates are false. The effect set of the above causal effect in that state consists of 1) the creation of one new domain element, say $\_p$, and 2) the caused atoms $\mathrm{Pack}(\_p)$, $\mathrm{Cont}(\_p, \mathrm{MyMail})$, $\mathrm{OnCh}(\_p, 1)$ and $\mathrm{Received}(\_p, 7)$, where instead of 7, we could have chosen any number greater than zero. Next, it continues, and in every step t, before receiving the package, an extra atom $\mathrm{OnCh}(p, t+1)$ is caused. Finally, in the seventh step, no more atoms are caused; the causal process ends. The final state is a model of the causal theory.

3 Semantics of Causal Theories

In this section we formalise the above intuitions. We start by formalising the effect set of a CEE without **New**-expressions, and introduce **New**-expressions afterwards. We conclude with a discussion on causal processes where preconditions of effects contain endogenous predicates and different such rules can affect each other's preconditions, i.e., where recursion is present.

3.1 Basic CEEs

In this subsection, we assume that all CEEs are **New**-free, i.e., they have no subexpressions of the form $\mathbf{New}\, x : C'$.

Definition 3.1. *Let C be a CEE and $\mathcal{I}$ a structure. We define the* possible effect sets *of C in $\mathcal{I}$ by induction:*

- *If C is $P(\overline{t})$, then the only possible effect set of C is the singleton $\{P(\overline{t}^{\mathcal{I}})\}$.*
- *If C is $C_1\, \mathbf{And}\, C_2$, then a set $V = V_1 \cup V_2$ is a possible effect set of C if V_1 is a possible effect set of C_1 and V_2 is a possible effect set of C_2.*
- *If C is $C_1\, \mathbf{Or}\, C_2$, then a set V is a possible effect set of C if V is a possible effect set of C_1 or of C_2.*
- *If C is $C' \leftarrow \varphi$, then a set V is a possible effect set of C if either $\varphi^{\mathcal{I}}$ is false and $V = \emptyset$ or $\varphi^{\mathcal{I}}$ is true and V is a possible effect set of C'.*
- *If C is $\mathbf{Select}\, x[\varphi] : C'$, then a set V is a possible effect set of C if there exists a $d \in D^{\mathcal{I}}$ such that $\mathcal{I}[x : d] \models \varphi$ and V is a possible effect set of C' in $\mathcal{I}[x : d]$.*
- *If C is $\mathbf{All}\, x[\varphi] : C'$, let $S = \{d \in D^{\mathcal{I}} \mid \mathcal{I}[x : d] \models \varphi\}$, then a set V is a possible effect set of C if V is the union of sets $(V_d)_{d \in S}$ such that each V_d is a possible effect set of C' in $\mathcal{I}[x : d]$.*

As is evident from this definition, a CEE C may have different possible effect sets in a structure, caused by non-deterministic choice on instances of **Or**- and **Select**-subexpressions. In the context of a causal process, it is important that, once such a non-deterministic choice has been made, it remains fixed throughout the entire sequence of states. For instance, if C contains $\mathbf{All}\, x[R(x)] : P(x)\, \mathbf{Or}\, Q(x)$, then for each domain element d the process chooses either $P(d)$ or $Q(d)$ and it sticks to this choice during the entire process. To enforce this, we introduce the following concepts.

Definition 3.2. *Let Δ be a causal theory; we associate a parse-tree with Δ. An* occurrence *of a CEE C in Δ is a node in the parse tree of Δ labelled with C. The* variable context *of an occurrence of a CEE C in Δ is the sequence of quantified variables that occur on the path from Δ to C in the parse-tree of Δ. If $\overline{x}$ is the variable context of C in Δ, we denote C as $C\langle\overline{x}\rangle$ and the length of $\overline{x}$ as n_C.*

For example, the variable context of $P(x)$ in $\mathbf{All}\, u[R(u)] : \mathbf{Select}\, y[Q(y)] : \mathbf{All}\, x[Q(x)] : P(x)$ is $[u, y, x]$.

Instances of an occurrence $C\langle\overline{x}\rangle$ correspond to assignments $\overline{d}$ of domain elements to $\overline{x}$.

Definition 3.3. *Let Δ be a causal theory and D a set. A* Δ-selection *ζ in D consists of*

- *for every occurrence C of a $\mathbf{Select}$-expression in Δ, a function $\zeta_C^{sel} : D^{n_C} \to D$,*
- *for every occurrence C of a $\mathbf{Or}$-expression in Δ, a function $\zeta_C^{or} : D^{n_C} \to \{1, 2\}$.*

Both **Or** and **Select**-expressions force the causal process to make a non-deterministic choice. The results of these choices can then be recorded in a Δ-selection ζ. For instance, let $C\langle y\rangle = \mathbf{Select}\, x[\varphi] : P(x, y)$, and $e = \zeta_C^{sel}(d)$. Then the effect set of the instance of C assigning d to y is $\{P(e, d)\}$. For $C\langle y\rangle = A\, \mathbf{Or}\, B$, the causal process will cause C_1 if $\zeta_C^{or}(y^{\mathcal{I}}) = 1$, and C_2 otherwise.[1]

Definition 3.4. *Let Δ be a CEE and $\mathcal{I}$ a structure. Suppose ζ is a Δ-selection in $D^{\mathcal{I}}$. Let $C\langle\overline{y}\rangle$ be an occurrence of a CEE in Δ. The* (actual) effect set of C *with respect to $\mathcal{I}$ and ζ (denoted by $\mathrm{eff}_{\mathcal{I},\zeta}(C)$) is a set of domain atoms, defined recursively as follows:*

- *If C is $P(\overline{t})$, then $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \{P(\overline{t}^{\mathcal{I}})\}$,*
- *if C is $C_1\, \mathbf{And}\, C_2$, then*
  $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \mathrm{eff}_{\mathcal{I},\zeta}(C_1) \cup \mathrm{eff}_{\mathcal{I},\zeta}(C_2)$,
- *if C is $C' \leftarrow \varphi$, then $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \mathrm{eff}_{\mathcal{I},\zeta}(C')$ if $\mathcal{I} \models \varphi$, and $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \emptyset$ otherwise,*
- *if C is $\mathbf{All}\, x[\varphi] : C'$, then*
  $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \bigcup\{\mathrm{eff}_{\mathcal{I}[x:d],\zeta}(C') \mid d \in D^{\mathcal{I}} \wedge \mathcal{I}[x : d] \models \varphi\}$,
- *if $C\langle\overline{y}\rangle$ is $C_1\, \mathbf{Or}\, C_2$, then*
  - $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \mathrm{eff}_{\mathcal{I},\zeta}(C_1)$ *if* $\zeta_C^{or}(\overline{y}^{\mathcal{I}}) = 1$,
  - *and* $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \mathrm{eff}_{\mathcal{I},\zeta}(C_2)$ *otherwise*
- *if $C\langle\overline{y}\rangle$ is $\mathbf{Select}\, x[\varphi] : C'$, let $\mathcal{I}' = \mathcal{I}[x : \zeta_C^{sel}(\overline{y}^{\mathcal{I}})]$,*
  - *then* $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \mathrm{eff}_{\mathcal{I}',\zeta}(C')$ *if* $\mathcal{I}' \models \varphi$,
  - *and* $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \emptyset$ *otherwise.*

Contrary to what the terminology might suggest, an effect set (as defined above) is not necessarily also a possible effect set (as defined in Definition 3.1). Whether this is the case depends on the Δ-selection ζ, and, in particular, on whether it makes correct choices for all of the **Select**-CEEs. To be more precise, if, for each CEE of the form $\mathbf{Select}\, x[\varphi] : C'$, ζ chooses only values for x that satisfy φ, then the effect set will be a possible effect set; otherwise it will not. In Definition 3.8 below, we will define our semantics in such a way that only appropriate Δ-selections are allowed.

3.2 Object-Creating CEEs

A possible effect set of a **New**-expression can contain an atom $\mathcal{U}(d)$, with $\mathcal{U}$ an auxiliary unary predicate symbol not in Σ; such an atom means that d is a "new" element.

Definition 3.1 (continued). *Let C be a CEE and $\mathcal{I}$ a structure. We extend the definition of* possible effect sets *to* **New***-expressions as follows:*

- *If C is $\mathbf{New}\, x : C'$, then a set $V = V' \cup \{\mathcal{U}(d)\}$ is a possible effect set of C if V' is a possible effect set of C' in $\mathcal{I}[[x : d]]$.[2] We call d the element created by C in $\mathcal{I}$.*

Informally, it is clear that different **New**-expressions should create different elements. Therefore, we will demand from a possible effect set that this is indeed the case and we will define a Δ-selection accordingly. However, it does not guarantee that the elements are really "new": we do not exclude the possibility that a created element d already exists in $\mathcal{I}$. This is a technical detail: during a causal process, at

[1] This notion of selection is reminiscent of the way the Hilbert operator ϵ is defined in (Ackermann 1925).

[2] We use $\mathcal{I}[[x : d]]$ for the structure with domain $D^{\mathcal{I}} \cup \{d\}$ equal to $\mathcal{I}$ except interpreting x by d.

some point when an expression $\mathbf{New}\, x : C'$ fires, it creates a new element. At later points in time we might need to re-evaluate the CEE because one of its subexpressions fires. In that case, a Δ-selection remembers which the created element n was and $C'(n)$ is caused for that specific n.

Definition 3.3 (continued). *In addition, a* Δ-selection *in* D *consists of*

- *for every occurrence* $C\langle\overline{y}\rangle$ *of a* $\mathbf{New}$*-expression in* Δ*, an injective partial function* $\zeta^n_C : D^{n_C} \to D$*,*

such that furthermore the images of all functions ζ^n_C *are disjoint (i.e., such that every domain element can only be created once).*

A remark concerning Definition 3.3 is that selection functions for $\mathbf{New}$-expressions are partial. This is needed because otherwise, their injectivity would imply that D should be infinite in almost all cases. Often, not all choices are relevant. Consider $(\mathbf{New}\, y[\mathbf{t}] : P(y))\, \mathbf{Or}\, (\mathbf{New}\, x[\mathbf{t}] : Q(x))$. Whichever choice we make for the $\mathbf{Or}$-expression, one of the two selections for $\mathbf{New}$-expressions is redundant. Intuitively, a Δ-selection should be defined only on those instances of $\mathbf{New}$-expressions where its effect set matters. If $\overline{d}$ has no image under ζ^n_C, we say that $\zeta^n_C(\overline{d})$ does not denote.

In order to incorporate newly created elements in the effect set of a CEE, we cannot simply use a Δ-selection in $D^{\mathcal{I}}$, as this way it would never be possible to actually create new objects. On the other hand, when we allow Δ-selections to range over a larger domain, it might be possible that the functions for $\mathbf{Select}$-expressions choose something outside $D^{\mathcal{I}}$. Therefore, we introduce a new concept that intuitively says when a Δ-selection is a good Δ-selection relative to $\mathcal{I}$.

Definition 3.5. *Let* ζ *be a* Δ*-selection in* D*; we call* ζ $\mathcal{I}$-compatible *if all selection functions* ζ^{sel}_C *map into* $D^{\mathcal{I}}$*.*

Definition 3.4 (continued). *Suppose* ζ *is an* $\mathcal{I}$*-compatible* Δ*-selection in some* $D \supseteq D^{\mathcal{I}}$*.*

- *if* $C\langle\overline{y}\rangle$ *is* $\mathbf{New}\, x : C'$*, then*
  - $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \emptyset$ *if* $\zeta^n_C(\overline{y}^{\mathcal{I}})$ *does not denote,*
  - *otherwise, let* $e = \zeta^n_C(\overline{y}^{\mathcal{I}})$*, then* $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \{\mathcal{U}(e))\} \cup \mathrm{eff}_{\mathcal{I}[[x:e]],\zeta}(C')$.

3.3 Recursion in Causal Theories

A Δ-selection in D specifies which elements are created, namely those that are in the image of one of the ζ^n_C functions. We define the set ζ^{in} of *initial elements of* ζ as the subset of D consisting of the non-created elements.

From now on, let Δ be a causal theory, $\mathcal{I}$ a fixed Σ-structure and ζ a Δ-selection in $D^{\mathcal{I}}$ such that for each constant σ, $\sigma^{\mathcal{I}} \in \zeta^{in}$. With $L^{\Sigma}_{\mathcal{I},\zeta}$ we denote the set of all Σ-structures J with $\zeta^{in} \subseteq D^J \subseteq D^{\mathcal{I}}$ such that for all exogenous symbols σ: $\sigma^J = \sigma^{\mathcal{I}}|_{D^J}$. Thus $L^{\Sigma}_{\mathcal{I},\zeta}$ are the structures with smaller domain than $\mathcal{I}$, at least containing all initial elements, such that $\mathcal{I}$ and J agree on the exogenous symbols of Δ. We define a lattice order $\leq$ on $L^{\Sigma}_{\mathcal{I},\zeta}$ as follows: $J \leq J'$ if $D^J \subseteq D^{J'}$ and for all predicates P, $P^J \subseteq P^{J'}$. The least element $\bot$ in this order is the structure with domain ζ^{in} interpreting all endogenous predicates as the empty set.

Given a Δ-selection ζ in $D^{\mathcal{I}}$, we can restrict ζ to be J-compatible by restricting the ζ^{sel}_C functions to D^J. We denote this restriction by $\zeta|_J$.

Definition 3.6. *We define the* immediate causality operator O_ζ *in* $L^{\Sigma}_{\mathcal{I},\zeta}$ *as the operator that sends each structure* J *to a structure* $J' \in L^{\Sigma}_{\mathcal{I},\zeta}$ *such that (using* ζ' *for* $\zeta|_J$*):*

- $d \in D^{J'}$ *if and only if* $d \in \zeta^{in}$ *or* $\mathcal{U}(d) \in \mathrm{eff}_{J,\zeta'}(\Delta)$*,*
- *for endogenous symbols* P*,* $\overline{d} \in P^{J'}$ *if and only if* $P(\overline{d}) \in \mathrm{eff}_{J,\zeta'}(\Delta)$*.*

The operator O_ζ applies the updates described by Δ to a structure, thus resulting in a new structure. Applying this operator iteratively, starting from the initial state $\bot$, yields a process similar to (4). If O_ζ is monotone, then the sequence constructed in this way will be such that caused atoms are never undone. In other words, once an atom is caused, it remains true in the rest of the process. This property ensures that the sequence converges. Monotonicity of O_ζ holds in many cases, as the following theorem illustrates.

Theorem 3.7. O_ζ *is a monotone lattice operator for every* ζ *if* Δ *satisfies one of the following conditions:*

- Δ *does not contain* $\mathbf{New}$*-expressions and no endogenous predicates occur in the scope of a negation in first-order formulas*[3] *in* Δ*, or*
- *all first-order formulas*[3] *in* Δ *are positive first-order formulas without universal quantification* $\forall$*.*

Idea of the proof. The above conditions guarantee that more qualifications of causal effects will hold in larger structures. If more qualifications hold, O_ζ derives more domain atoms, and hence O_ζ is monotone. □

In case of a monotone operator, it is easy to define when a structure is a model of Δ, or intuitively, when a structure can be constructed using the causal laws in Δ.

Definition 3.8. *Let* Δ *be such that one of the two conditions in Theorem 3.7 is satisfied. We say that* $\mathcal{I}$ *is a* model *of* Δ *(notation* $\mathcal{I} \models \Delta$*) if there exists a* Δ*-selection* ζ *such that* $\mathcal{I}$ *is the least fixpoint of* O_ζ*, and* $\mathrm{eff}_{\mathcal{I},\zeta}(\Delta)$ *is a possible effect set of* Δ *in* $\mathcal{I}$*.*

The intuitions behind this definition have already been explained in the context of CP-logic (Vennekens, Denecker, and Bruynooghe 2009). A model should be a fixpoint of Δ because of the principle of *sufficient causation*: if the precondition for a causal law is satisfied, then the event that it triggers must eventually happen. The fact that we demand that it is the *least* fixpoint comes from the principle of *universal causation*: all changes to the state of the domain must be triggered by a causal law whose precondition is satisfied. The fact that it is the least fixpoint also excludes deus-ex-machina effects, where an atom is caused based on its own truth, e.g., a rule $Q \leftarrow Q$ can never cause Q. The second condition, that $\mathrm{eff}_{\mathcal{I},\zeta}(\Delta)$ is a possible effect set, is a technical detail to exclude bad Δ-selections. This excludes situations where an instance of a non-deterministic CEE causes

[3] Qualifications of $\mathbf{All}$- and $\mathbf{Select}$-expressions and conditions of rule-expressions.

nothing because a selection function makes a bad choice—an element not satisfying the qualification—or (in the case of **New**-expressions) a selection function does not denote.

Definition 3.8 formalises the idea that each (instantiation of) a **New**-expression creates a unique different element that extends the domain. Uniqueness is guaranteed by the definition of Δ-selection; the least-fixpoint computation starts from a domain ζ^{in}, hence **New**-expressions really extend the domain.

Example 2.2 (continued). Let us return to the example of the American Lottery.

$$\mathbf{All}\, p[\mathrm{Applied}(p) \wedge \mathrm{PassedTest}(p)] : \mathrm{American}(p)$$
$$\mathbf{Select}\, p[\mathrm{Applied}(p)] : \mathrm{American}(p) \leftarrow \mathrm{Lottery}$$

Suppose a domain $\{a, b, c, d\}$ of persons is given. Let $\mathcal{I}$ denote a structure such that $\mathrm{Lottery}^{\mathcal{I}} = \mathbf{t}$, $\mathrm{Applied}^{\mathcal{I}} = \{a, b\}$ and $\mathrm{PassedTest}^{\mathcal{I}} = \mathrm{American}^{\mathcal{I}} = \emptyset$. The structure $\mathcal{I}$ is not an intended model of the causal theory since the lottery took place, but no-one won. With Δ-selection ζ such that $\zeta^{sel}_C() = c$ (with C being the **Select**-expression), $\mathcal{I}$ is the least fixpoint of O_ζ. The choice c for the **Select**-expression violates its qualification, hence $\mathrm{eff}_{\mathcal{I},\zeta}(\Delta)$ is not a possible effect set. We find that indeed $\mathcal{I}$ is not a model of Δ. Models of Δ are exactly those structures in which everyone who passed the test and one random (not necessarily different) person obtain the American nationality.

Example 3.9. Let us consider the creation of the natural numbers. This can be seen as a causal process in which every number causes one new number—its successor—to be created. We express this in C-log as follows:

$$\mathbf{New}\, x : (\mathrm{Nat}(x)\, \mathbf{And}\, \mathrm{Zero}(x))$$
$$\mathbf{All}\, x[\mathrm{Nat}(x)] : \mathbf{New}\, y : (\mathrm{Nat}(y)\, \mathbf{And}\, \mathrm{Succ}(x, y)).$$

Notice that for any ζ, O_ζ is monotone. All models $\mathcal{I}$ of Δ interpret Nat as the natural numbers (modulo isomorphism). The corresponding ζ has a nullary selection function $\zeta^n_{C_0}$ such that $\zeta^n_{C_0}() = 0$ (modulo isomorphism), and unary selection function $\zeta^n_{C_1}$ such that for each natural number n, $\zeta^n_{C_1}(n) = n+1$ for the above CEE-expressions respectively. The intermediate states $\mathcal{I}_i$ have domain $\zeta^{in} \cup [0, i-1]$, and interpret Nat as $[0, i-1]$. The limit structure is the least fixpoint of O_ζ, and $\mathrm{eff}_{\mathcal{I},\zeta}(\Delta)$ is indeed a possible effect set.

For any other structure $\mathcal{I}$ and corresponding ζ, i.e., for structures $\mathcal{I}$ not interpreting Nat as the natural numbers, $\mathrm{Nat}^{\mathcal{I}}$ is either finite in which case $\zeta^n_{C_1}$ is partial on $\mathrm{Nat}^{\mathcal{I}}$ and in this case, $\mathrm{eff}_{\mathcal{I},\zeta}(\Delta)$ is not a possible effect set, or $\mathrm{Nat}^{\mathcal{I}}$ contains more elements, in which case $\mathcal{I}$ is not the least fixpoint of O_ζ.

4 Semantics of General Causal Theories

The strategy described in Section 3.3 does not always work. When negation is present, O_ζ can derive unintended effects.

Example 2.2 (continued). We extend the example about American citizenship with two more laws: an American citizen obtains a green card (GCO stands for "Green Card Obtained") when he comes to an office; the office closes when every American has received his green card.

$$\mathbf{All}\, x[\mathrm{American}(x) \wedge \mathrm{ComeToOffice}(x)] : \mathrm{GCO}(x)$$
$$\mathrm{Close} \leftarrow (\forall x[\mathrm{American}(x)] : \mathrm{GCO}(x))$$

Especially the last rule is of interest to us because it contains a negative occurrence of $\mathrm{American}(x)$. If we consider the state in which all endogenous predicates are false, the effect set of the above CEE contains Close. However, we intended the office to stay open until all winners have their card.

The reason why in the above example things go wrong is because the process that causes x to become "American" is still ongoing when the last rule is executed. The condition in the last rule actually refers to final state; we can only execute it as soon as we know who the Americans are. We could fix this by introducing an order on causal effects, but doing so would strongly compromise the usability of the logic. Instead, we use an existing framework to define valid causation sequences for general causal theories. The idea is that we work with partial information: instead of applying an operator to a structure, we apply it to a partial structure containing information about the atoms that have been caused, and the atoms that might still be caused

4.1 Partial Structures with Partial Domain

We briefly summarise some concepts from three-valued logic and generalise them to allow open domains. A truth-value is one of the following: $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ (true, false and unknown). We define $\mathbf{f}^{-1} = \mathbf{t}, \mathbf{t}^{-1} = \mathbf{f}$ and $\mathbf{u}^{-1} = \mathbf{u}$. Two interesting partial orders are defined on the set of truth values: the precision order $\leq_p$, given by $\mathbf{u} \leq_p \mathbf{t}$ and $\mathbf{u} \leq_p \mathbf{f}$ and the truth order $\mathbf{f} \leq \mathbf{u} \leq \mathbf{t}$.

Definition 4.1. *A* partial set $\mathcal{S}$ *is a function from objects to truth values.*

We identify a partial set with a tuple $(\mathcal{S}_{ct}, \mathcal{S}_{pt})$ of two sets, where the *certainly true set* $\mathcal{S}_{ct}$ is $\{x \mid \mathcal{S}(x) = \mathbf{t}\}$ and the *possibly true set* $\mathcal{S}_{pt}$ is $\{x \mid \mathcal{S}(x) \neq \mathbf{f}\}$. The union, intersection, and subset-relation of partial sets are all defined pointwise. For a truth value v, we define the restriction of a partial set $\mathcal{S}$ to this truth-value, denoted $r(\mathcal{S}, v)$, as the partial set mapping x to $\min_{\leq}(\mathcal{S}(x), v)$.

A partial Σ-structure $\mathcal{I}$ consists of 1) a *domain* $D^{\mathcal{I}}$: a partial set of elements, and 2) a mapping associating a value to each symbol in Σ; for constants, this value is in $D^{\mathcal{I}}_{\mathrm{ct}}$, for predicate symbols this is a partial set $P^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. We often abuse notation and use the domain D as if it were a predicate. A partial structure $\mathcal{I}$ is *two-valued* if for all predicates P (including D), $P^{\mathcal{I}}_{\mathrm{ct}} = P^{\mathcal{I}}_{\mathrm{pt}}$. There is an obvious one-to-one correspondence between two-valued partial structures and structures. If $\mathcal{I}$ and $\mathcal{J}$ are two partial structures with the same interpretation for constants, we call $\mathcal{I}$ more precise than $\mathcal{J}$ ($\mathcal{I} \geq_p \mathcal{J}$) if for all its predicates P (including D), $P^{\mathcal{I}}_{\mathrm{ct}} \supseteq P^{\mathcal{J}}_{\mathrm{ct}}$ and $P^{\mathcal{I}}_{\mathrm{pt}} \subseteq P^{\mathcal{J}}_{\mathrm{pt}}$.

We now define the value of an FO formula in a partial structure as an extension of the Kleene valuation (Kleene 1938). Intuitively, the value of a quantification $\forall x : \varphi$, is the value of the restricted quantification $\forall x[D^{\mathcal{I}}(x)] : \varphi$, i.e., if there is enough information to derive that all elements that

might be in the domain, satisfy φ, this formula is true; if there is enough information to conclude that for one element in the domain, φ does not hold, this formula is false; otherwise, its value is unknown.

Definition 4.2. *Given a partial structure $\mathcal{I}$, we define the Kleene valuation ($Kl_{\mathcal{I}}$) inductively based on the Kleene truth tables:*

- $Kl_{\mathcal{I}}(P(\bar{t})) = P^{\mathcal{I}}(\bar{t}^{\mathcal{I}})$,
- $Kl_{\mathcal{I}}(\neg\varphi) = (Kl_{\mathcal{I}}(\varphi))^{-1}$
- $Kl_{\mathcal{I}}(\varphi \wedge \psi) = \min_{\leq}(Kl_{\mathcal{I}}(\varphi), Kl_{\mathcal{I}}(\psi))$
- $Kl_{\mathcal{I}}(\varphi \vee \psi) = \max_{\leq}(Kl_{\mathcal{I}}(\varphi), Kl_{\mathcal{I}}(\psi))$
- $Kl_{\mathcal{I}}(\forall x : \varphi) = \min_{\leq}\left\{\max(D^{\mathcal{I}}(d)^{-1}, Kl_{\mathcal{I}[x:d]}(\varphi))\right\}$
- $Kl_{\mathcal{I}}(\exists x : \varphi) = \max_{\leq}\left\{\min(D^{\mathcal{I}}(d), Kl_{\mathcal{I}[x:d]}(\varphi))\right\}$

4.2 Causal Theories and Partial Structures

The effect set of a CEE in a partial structure is a partial set: it contains information on everything that is caused and everything that might be caused. Intuitively, it differs from Definition 3.4 by evaluating all quantifications relative to $D^{\mathcal{I}}$.

Definition 4.3. *Let Δ be a CEE and $\mathcal{I}$ a partial structure. Suppose ζ is an $\mathcal{I}$-compatible Δ-selection. Let C be an occurrence of a CEE in Δ. The* effect set of *C with respect to $\mathcal{I}$ and ζ is a partial set of domain atoms, defined recursively:*

- *If C is $P(\bar{t})$, then* $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \{P(\bar{t}^{\mathcal{I}})\}$,
- *if C is C_1* **And** *C_2, then*
  $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \mathrm{eff}_{\mathcal{I},\zeta}(C_1) \cup \mathrm{eff}_{\mathcal{I},\zeta}(C_2)$,
- *if C is a rule-expression $C' \leftarrow \varphi$, then*
  $\mathrm{eff}_{\mathcal{I},\zeta}(C) = r(\mathrm{eff}_{\mathcal{I},\zeta}(C'), Kl_{\mathcal{I}}(\varphi))$,
- *if C is* **All** *$x[\varphi] : C'$, then*
  $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \bigcup\{r(\mathrm{eff}_{\mathcal{I}',\zeta}(C'), \min_{\leq}(D^{\mathcal{I}}(d), Kl_{\mathcal{I}'}(\varphi))) \mid d \in D^{\mathcal{I}}_{\mathrm{pt}}$ *and* $\mathcal{I}' = \mathcal{I}[x : d]\}$
- *if $C\langle\bar{y}\rangle$ is C_1* **Or** *C_2, then*
  - $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \mathrm{eff}_{\mathcal{I},\zeta}(C_1)$ *if* $\zeta^{or}_C(\bar{y}^{\mathcal{I}}) = 1$,
  - *and* $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \mathrm{eff}_{\mathcal{I},\zeta}(C_2)$ *otherwise*
- *if $C\langle\bar{y}\rangle$ is* **Select** *$x[\varphi] : C'$, let $e = \zeta^{sel}_C(\bar{y}^{\mathcal{I}})$, $\mathcal{I}' = \mathcal{I}[x : e]$ and $v = \min_{\leq}(D^{\mathcal{I}}(e), Kl_{\mathcal{I}'}(\varphi))$, then*
  $\mathrm{eff}_{\mathcal{I},\zeta}(C) = r(\mathrm{eff}_{\mathcal{I},\zeta}(C'), v)$,
- *if $C\langle\bar{y}\rangle$ is* **New** *$x : C'$, then*
  - $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \emptyset$ *if* $\zeta^{n}_C(\bar{y}^{\mathcal{I}})$ *does not denote,*
  - *and* $\mathrm{eff}_{\mathcal{I},\zeta}(C) = \{\mathcal{U}(\zeta^{n}_C(\bar{y}^{\mathcal{I}}))\} \cup \mathrm{eff}_{\mathcal{I}',\zeta}(C')$, *where* $\mathcal{I}' = \mathcal{I}[[x : \zeta^{n}_C(\bar{y}^{\mathcal{I}})]]$ *otherwise,*

We now define the partial immediate causality operator, a three-valued variant of O_ζ similar to the well-known Fitting operator for logic programs (Fitting 1985).

Definition 4.4. *The* partial immediate causality operator A_ζ *sends each partial structure J to a partial structure J' with the same interpretation for exogenous symbols and such that*

- $D^{J'}(d) = \mathbf{t}$ *if* $d \in \zeta^{in}$ *and* $D^{J'}(d) = \mathrm{eff}_{J,\zeta}(\Delta)(\mathcal{U}(d))$ *otherwise, and*
- *for endogenous symbols P,* $P(\bar{d})^{J'} = \mathrm{eff}_{J,\zeta}(\Delta)(P(\bar{d}))$.

Simply iterating the above operator does not possess the nice properties we found in Section 3.3. Consider the bike example from the introduction. If P is false, we would expect that none of the gears starts turning. However, iteratively applying A_ζ, starting from the initial state, where $T(B) = T(S) = \mathbf{u}$, also ends in this state, instead of ending in the structure where both atoms are false.

Luckily, such principles have been studied intensively in the field of Approximation Fixpoint Theory.

4.3 Approximation Fixpoint Theory (AFT)

In this section, we summarise the basic concepts from Approximation Fixpoint Theory (Denecker, Marek, and Truszczyński 2000). From now on, let $(L, \leq)$ be a *complete lattice*, i.e., $\leq$ is a partial order on L and every subset S of L has a *least upper bound* $\mathrm{lub}_{\leq}(S)$ and a *greatest lower bound* $\mathrm{glb}_{\leq}(S)$. We write $\bot$ and $\top$ for $\mathrm{glb}_{\leq}(L)$ and $\mathrm{lub}_{\leq}(L)$ respectively. Pairs $(x, y) \in L^2$ are used to approximate all elements in the interval $[x, y] = \{z \mid x \leq z \wedge z \leq y\}$. *Projections* on such pairs are defined as usual: $(x, y)_1 = x$ and $(x, y)_2 = y$. The *precision ordering* on L^2 is defined as $(x, y) \leq_p (u, v)$ if $x \leq u$ and $v \leq y$, i.e., if (x, y) approximates all elements approximated by (u, v), or in other words if $[x, y] \subseteq [u, v]$. We call $(x, y) \in L^2$ *consistent* if it approximates at least one element, and use L^c to denote all consistent elements. Elements $(x, x) \in L^c$ are called *exact*.

An operator $O : L \to L$ is *monotone* if it respects $\leq$, i.e., if $x \leq y$ implies $O(x) \leq O(y)$. An element $x \in L$ is a *fixpoint* of O if $O(x) = x$. An operator $A : L^2 \to L^2$ is an *approximator* of O if it is $\leq_p$-monotone, internal in L^c (i.e., it maps L^c into L^c) and has the property that for all x, $O(x) \in [x', y']$, where $(x', y') = A(x, x)$. (Denecker and Vennekens 2007) defines the notion of *well-founded inductions* of such an approximator. This concept is based on the following intuitions. A well-founded induction of A is a process that uses A to derive more and more information (i.e., to grow in $\leq_p$). For CEEs, this will be in accordance with the principle of sufficient causation. At the same time, a well-founded induction tries to minimize the upper bounds $(x, y)_2$. In the context of CEEs, this means that it tries to minimize the number of new elements created, as well as the truth of endogenous predicates. In this way, it incorporates the principle of universal causation.

Definition 4.5. *An* A-refinement *of (x, y) is a pair $(x', y') \in L^2$ satisfying one of the following two conditions*

- $(x, y) \leq_p (x', y') \leq_p A(x, y)$,
- *or $x' = x$ and y' satisfies* $A(x, y')_2 \leq y'$.

Definition 4.6. *A* well-founded induction *of A in (x, y) is a sequence $\langle(x_\xi, y_\xi)\rangle_{\xi \leq \alpha}$ with α an ordinal such that*

- $(x_0, y_0) = (x, y)$;
- $(x_{\xi+1}, y_{\xi+1})$ *is an A-refinement of (x_ξ, y_ξ), for all $\xi < \alpha$;*
- $(x_\lambda, y_\lambda) = \mathrm{lub}_{\leq_p}(\{(x_\xi, y_\xi) \mid \xi < \lambda\})$, *for each limit ordinal $\lambda \leq \alpha$.*

A well-founded induction is terminal *if its limit (x_α, y_α) has no strict A-refinements.*

For a given approximator A and a pair (x, y), there are many different terminal well-founded inductions of A in (x, y). However, as proven in (Denecker and Vennekens 2007), they all have the same limit, called the *A-well-founded point* of (x, y). The *well-founded point of A,* is the A-well-founded point of $(\bot, \top)$. In (Denecker, Marek, and Truszczyński 2004) it was shown that it suffices to define an approximator only on elements of L^c in order to uniquely characterize its well-founded point.

4.4 Models of General Causal Theories

Now that we recalled the basics of AFT, we can apply this theory to C-log. First, we note that $(L^{\Sigma}_{\mathcal{I},\zeta}, \leq)$ is indeed a lattice. Furthermore the consistent part of $(L^{\Sigma}_{\mathcal{I},\zeta})^2$ is exactly the set of partial structures under the precision order. The following lemma shows that AFT is indeed applicable.

Lemma 4.7. *A_ζ is an approximator of O_ζ.*

As explained above, the notion of a well-founded induction captures—in a general, algebraic way—the intuitions behind both the principles of sufficient and universal causation. This motivates the following definition.

Definition 4.8. *Let Δ be any causal theory. We say that structure $\mathcal{I}$ is a* model *of Δ (notation $\mathcal{I} \models \Delta$) if there exists a Δ-selection ζ such that ($\mathcal{I}$,$\mathcal{I}$) is the well-founded point of A_ζ, and $\mathrm{eff}_{\mathcal{I},\zeta}(\Delta)$ is a possible effect set of Δ in $\mathcal{I}$.*

For monotone operators O_ζ, the well-founded point of A_ζ is the least fixpoint of O_ζ. Hence, this definitions generalizes Definition 3.8. When applying this definition to the extended American lottery example, we see that well-founded inductions correctly model the desired behaviour. In the first step of a well-founded induction starting from the initial state, some atoms $\mathrm{American}(d)$ are derived; the other predicates remain unchanged. In the second step, rightful Americans get their green card, and Close still remains unknown. Only in the last phase of the induction, Close is derived.

The above example shows that even though no explicit ordering of execution of causal effects is given, AFT manages to guess the correct ordering. As shown in (Vennekens, Denecker, and Bruynooghe 2009), the well-founded induction process essentially introduces an assumption of *temporal precedence*, whereby all CEEs that might cause a certain effect are assumed to take place before this effect itself. While this assumption is not necessarily always correct, it has been shown to be typically satisfied.

5 FO(C-log): Integrating FO and C-log

First-order logic and C-log have a straightforward integration, FO(C-log). Theories in this logic are sets of FO sentences and CEEs. A model of such a theory is a structure that satisfies each of its expressions (each of its CEEs and formulas). An illustration is the mail protocol from Example 2.3, which we can extend with the "observation" that at at some time point, two packages are on the channel:

$$\exists t, p_1, p_2 : \mathrm{OnCh}(p_1, t) \wedge \mathrm{OnCh}(p_2, t) \wedge p_1 \neq p_2.$$

Models of this theory represent states of affairs where at least once two packages are on the channel simultaneously. This entirely differs from $\mathbf{And}$-conjoining our CEE with

$$\mathbf{Select}\, t[\mathbf{t}] : \mathbf{Select}\, p_1[\mathbf{t}] : \mathbf{Select}\, p_2[p_1 \neq p_2] : \\ \mathrm{OnCh}(p_1, t)\ \mathbf{And}\ \mathrm{OnCh}(p_2, t).$$

The resulting CEE would have unintended models in which two packages suddenly appear on the channel for no reason.

In FO(C-log), $\mathbf{New}$-expressions can be simulated with $\mathbf{Select}$-expressions together with FO axioms expressing the unicity of the newly "created" objects. E.g., $\mathbf{New}\, x : P(x, a)\ \mathbf{And}\ \mathbf{New}\, x : Q(x)$ is simulated by introducing auxiliary unary predicates N_1 and N_2 that identify the objects created by the expressions and writing:

$$\left\{ \begin{array}{c} (\mathbf{Select}\, x[\mathbf{t}] : (N_1(x)\ \mathbf{And}\ P(x, a)))\mathbf{And} \\ \mathbf{Select}\, x[\mathbf{t}] : (N_2(x)\ \mathbf{And}\ Q(x)) \end{array} \right\} \\ \forall x : \neg(N_1(x) \wedge N_2(x))$$

It is clear that $\mathbf{New}$-expressions are more natural and more modular than this simulation.

Despite the syntactical correspondence between CEEs and FO formulas ($\mathbf{And}$ corresponds to $\wedge$, $\mathbf{All}$ to $\forall$, ...), it is obvious that they have an entirely different meaning, and that both are useful. This is why we chose to introduce new connectives rather than overloading the ones of FO. The logic FO(C-log) has further interesting extensions, e.g., by adding aggregates in FO formulas, including in qualifications and conditions of CEEs.

6 Comparison and Future Work

Due to its simple recursive syntax, C-log is a very general logic that generalises several existing logics and shows overlaps with many others in different areas of computational logic. C-log is an extension of (the non-probabilistic version of) CP-logic. FO(C-log) is an extension of the logic FO(ID) as defined in (Denecker and Ternovska 2008). An FO(ID) theory is a set of FO sentences and inductive definitions (ID), which are sets of rules of the form

$$\forall \overline{x} : P(\bar{t}) \leftarrow \varphi,$$

where φ is an FO formula. Such a rule corresponds to a CEE

$$\mathbf{All}\, \overline{x}[\varphi] : P(\bar{t})$$

or equivalently,

$$\mathbf{All}\, \overline{x}[\mathbf{t}] : (P(\bar{t}) \leftarrow \varphi)$$

and a definition corresponds to the $\mathbf{And}$-conjunction of its rules. (Denecker, Theseider-Dupré, and Van Belleghem 1998) already pointed to the correspondence between causality and inductive definitions and exploited it for solving the causal *ramification problem* of temporal reasoning (McCarthy and Hayes 1969). The CEEs presented here can be seen as a non-deterministic extension of inductive definitions with an informal semantics based on causal processes.

C-log shows similarity to extensions of disjunctive logic programming (DLP) such as DLP with existential quantification in rule heads (You, Zhang, and Zhang 2013) and the

stable semantics for FO as defined in (Ferraris, Lee, and Lifschitz 2011). However, there is an important semantical difference. Suppose we want to express Example 2.2, where all people passing a test and one random person are given the American nationality. The E-disjunctive program

$$\exists X : american(X) \text{:-} lottery$$
$$\forall X : american(X) \text{:-} passtest(X)$$

is similar to

$$(\mathbf{Select}\ x[\mathbf{t}] : american(x)) \leftarrow lottery$$
$$\mathbf{All}\ x[passtest(x)] : american(x)$$

Semantically, the first imposes a minimality condition: the lottery is always won by a person succeeding the test, if there exists one. On the other hand, in C-log the two rules execute independently, and models might not be minimal. In this example, it is the latter that is intended. We believe that one advantage of C-log is its clear causal informal semantics. On the other hand, there are ways to simulate the causal semantics and the **New** operator of C-log in E-disjunctive programs while it follows from complexity arguments that not all E-disjunctive programs can be expressed in C-log.

Other semantics than the stable semantics for DLP have been developed. For example in (Brass and Dix 1996), D-WFS, a well-founded semantics was proposed. This semantics has the property that if a program contains two identical lines, one of them can be removed. However, in our context, a duplicate effect means that a same causal effect happens twice (maybe for different reasons), independently, and hence different choices might be made in each of these rules.

The logic of cause and change of (McCain and Turner 1996) differs with C-log in several important aspects; in McCain & Turner's logic both true and false atoms need a cause. In C-log on the other hand, endogenous predicates can be false (the default value) without reason but can only be true (the deviant value) if caused. Moreover, we rule out unfounded "cyclic" causation. For instance, if Pedal is false, in C-log, $\mathrm{Turn(BigGear)}$ and $\mathrm{Turn(SmallGear)}$ are false but in McCain and Turner's logic they may be true and caused by each other. We call this "spontaneous generation" and do not admit it in C-log.

We find operators similar to those of C-log in several other formalisms. For example, **Select**-, **All**-, **Or**- and rule-expressions are present in the subformalism of the language Event-B that serves to specify effects of actions (Abrial 2010). The **New** operator is found in various other rule based paradigms, for example in Business Rules systems (Business Rules Group 2000). The JBoss manual (Browne 2009) contains the following rule:

```
when Order( customer == null )
then
 insertLogical(new ValidationResult(
   validation.customer.missing ));
```

meaning that if an order is created without customer, a new *ValidationResult* is created with the message that the customer is missing. This can be translated to C-log as follows:

$$\mathbf{All}\ y[\mathrm{Order}(y) \wedge \mathrm{NoCustumer}(y)] :$$
$$\mathbf{New}\ x : \mathrm{ValidationR}(x)\ \mathbf{And}\ \mathrm{Message}(x, \text{"}\ldots\text{"}).$$

Another field in which related language constructions have been developed is the field of deductive databases. In (Abiteboul and Vianu 1991), various extensions of Datalog are considered, resulting in non-deterministic semantics for queries and updates. One of the studied extensions is object creation. Such an extension is present in the LogicBlox system (Green, Aref, and Karvounarakis 2012). An example from the latter paper is the rule:

$$President(p), presidentOf[c] = p \leftarrow Country(c)$$

which means that for every country c, a new (anonymous) "derived entity" of type $President$ is created. Such rules with implicit existentially quantified head variables correspond with **New**-expressions in C-log.

Other Datalog extensions with other forms of object creation exist. For example (Van den Bussche and Paredaens 1995) discusses a version with creation of sets and compares its expressivity with simple object creation.

Non-deterministic choices have been studied intensively in the context of deductive databases. In (Krishnamurthy and Naqvi 1988), a non-deterministic choice in Datalog was introduced. This choice was *static*: choice models are constructed in three steps. First, models are calculated while ignoring choices (choosing everything); second, this model is used to select a number of choices for all occurrences of *choice goals* and third, models are recalculated with respect to these choice goals. In (Saccà and Zaniolo 1990; Giannotti et al. 1991), it is argued that static choices, do not behave well in the presence of recursion; hence *dynamic* choices were introduced. In (Saccà and Zaniolo 1990), stable models provide a model-theoretic description of these dynamic choices. In (Weidong and Jinghong 1996) an alternative choice principle on predicates P is introduced in which the values in certain argument positions in the tuples of P are non-deterministically chosen in function of the values at the other argument positions. The semantics of that logic is based on the well-founded semantics; this choice principle is very different from the principle in C-log. Compared to these, C-log resembles most the language of (Saccà and Zaniolo 1990); the difference is that C-log supports a recursive syntax and is based on the well-founded semantics, whereas (Saccà and Zaniolo 1990) uses stable semantics.

The above similarities suggest that C-log is a promising language to study and unify many existing logical paradigms and to provide a clear informal semantics for them. An in-depth semantical analysis of the exact relationship between C-log and the languages described above is an interesting topic for future work. Another research challenge is extending FO(C-log) with types, function symbols, arithmetic, etc. in order to make it useful as a KR-language. We need to study the complexity of various inference tasks in C-log, and develop and implement algorithms for these various tasks. Another research question is to add probabilities to C-log to obtain an extension of the probabilistic CP-logic, and possibly also of other related logics such as BLOG (Milch et al. 2005) and P-Log (Baral, Gelfond, and Rushton 2004).

References

Abiteboul, S., and Vianu, V. 1991. Datalog extensions for database queries and updates. *J. Comput. Syst. Sci.* 43(1):62–124.

Abrial, J.-R. 2010. *Modeling in Event-B - System and Software Engineering*. Cambridge University Press.

Ackermann, W. 1925. Begründung des "tertium non datur" mittels der Hilbertschen theorie der widerspruchsfreiheit. *Mathematische Annalen* 93(1):1–36.

Baral, C.; Gelfond, M.; and Rushton, N. 2004. Probabilistic reasoning with answer sets. In *Proc. Logic Programming and Non Monotonic Reasoning, LPNMR'04*, 21–33. Springer-Verlag.

Brass, S., and Dix, J. 1996. Characterizing D-WFS: Confluence and iterated GCWA. In Alferes, J. J.; Pereira, L. M.; and Orlowska, E., eds., *Logics in Artificial Intelligence*, volume 1126 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 268–283.

Browne, P. 2009. *JBoss Drools Business Rules*. From technologies to solutions. Packt Publishing, Limited.

Business Rules Group. 2000. Defining Business Rules $\sim$ What Are They Really? Technical report.

Cabalar, P. 2012. Causal logic programming. In Erdem, E.; Lee, J.; Lierler, Y.; and Pearce, D., eds., *Correct Reasoning*, volume 7265 of *Lecture Notes in Computer Science*. Springer. 102–116.

Denecker, M., and Ternovska, E. 2008. A logic of nonmonotone inductive definitions. *ACM Transactions on Computational Logic (TOCL)* 9(2):14:1–14:52.

Denecker, M., and Vennekens, J. 2007. Well-founded semantics and the algebraic theory of non-monotone inductive definitions. In Baral, C.; Brewka, G.; and Schlipf, J. S., eds., *LPNMR*, volume 4483 of *LNCS*, 84–96. Springer.

Denecker, M.; Marek, V. W.; and Truszczyński, M. 2000. Approximating operators, stable operators, well-founded fixpoints and applications in non-monotonic reasoning. In *Logic-based Artificial Intelligence*, The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Boston. 127–144.

Denecker, M.; Marek, V. W.; and Truszczyński, M. 2004. Ultimate approximation and its application in nonmonotonic knowledge representation systems. *Information and Computation* 192(1):84–121.

Denecker, M.; Theseider-Dupré, D.; and Van Belleghem, K. 1998. An inductive definition approach to ramifications. *Linkoping Electronic Articles in Computer and Information Science* 3(7):1–43.

Ferraris, P.; Lee, J.; and Lifschitz, V. 2011. Stable models and circumscription. *Artificial Intelligence* 175:236–263.

Fitting, M. 1985. A Kripke-Kleene semantics for logic programs. *The Journal of Logic Programming* 2(4):295 – 312.

Giannotti, F.; Pedreschi, D.; Saccà, D.; and Zaniolo, C. 1991. Non-determinism in deductive databases. In Delobel, C.; Kifer, M.; and Masunaga, Y., eds., *Deductive and Object-Oriented Databases*, volume 566 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 129–146.

Green, T. J.; Aref, M.; and Karvounarakis, G. 2012. Logicblox, platform and language: A tutorial. In Barceló, P., and Pichler, R., eds., *Datalog*, volume 7494 of *Lecture Notes in Computer Science*, 1–8. Springer.

Hall, N. 2004. Two concepts of causation. In *Causation and Counterfactuals*.

Hitchcock, C. 2007. Prevention, preemption, and the principle of sufficient reason. *Philosophical review* 116(4).

Kleene, S. C. 1938. On notation for ordinal numbers. *The Journal of Symbolic Logic* 3(4):pp. 150–155.

Krishnamurthy, R., and Naqvi, S. A. 1988. Nondeterministic choice in datalog. In *JCDKB*, 416–424.

McCain, N., and Turner, H. 1996. Causal theories of action and change. In *AAAI/IAAI*, 460–465. AAAI Press.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., and Michie, D., eds., *Machine Intelligence 4*. Edinburgh University Press. 463–502.

Milch, B.; Marthi, B.; Russell, S. J.; Sontag, D.; Ong, D. L.; and Kolobov, A. 2005. Blog: Probabilistic models with unknown objects. In Kaelbling, L. P., and Saffiotti, A., eds., *IJCAI*, 1352–1359. Professional Book Center.

Pearl, J. 2000. *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Preyer, G., and Peter, G. 2002. *Logical Form and Language*. Clarendon Press.

Saccà, D., and Zaniolo, C. 1990. Stable models and non-determinism in logic programs with negation. In *Proceedings of the Ninth ACM Symposium on Principles of Database Systems*, 205–217. ACM Press.

Van den Bussche, J., and Paredaens, J. 1995. The expressive power of complex values in object-based data models. *Information and Computation* 120:220–236.

Vennekens, J.; Denecker, M.; and Bruynooghe, M. 2009. CP-logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming* 9(3):245–308.

Vennekens, J.; Denecker, M.; and Bruynooghe, M. 2010. Embracing events in causal modelling: Interventions and counterfactuals in CP-logic. In Janhunen, T., and Niemela, I., eds., *Lecture Notes in Computer Science,*, 313–325. Springer.

Weidong, C., and Jinghong, Z. 1996. Nondeterminism through well-founded choice. *The Journal of Logic Programming* 26(3):285–309.

You, J.-H.; Zhang, H.; and Zhang, Y. 2013. Disjunctive logic programs with existential quantification in rule heads. *Theory and Practice of Logic Programming* 13:563–578.